Improving Quality and Speed of Range Data for Communication

Leonid V. Tsap
Center for Applied Scientific Computing
University of California Lawrence Livermore
National Laboratory, Livermore, CA 94551
tsap@llnl.gov

Min C. Shin
Department of Computer Science
University of North Carolina at Charlotte
Charlotte, NC 28223
mcshin@uncc.edu

Abstract [1]

Range data is very important in human-computer interaction applications. Although less costly, range acquisition and processing still presents a speed vs. data reliability tradeoff. This paper proposes a method that, given noisy and generally unreliable range data, can filter out erroneous information using range histograms. Then, using the resulting consistent data that has passed filters, this method limits the depth search space dynamically using motion history and its current state.

Experimental results demonstrate the success of the proposed algorithm. Using filtered range data, the algorithm correctly identified the hand involved in manipulation 99.8% of the time. Dynamic disparity adjustment produced a speedup of 60.2% over a static disparity range selection. An application to virtual reality navigation is also presented.

1 Introduction

1.1 Motivation: Why Range?

The usefulness of range data in interfaces based on human-computer communication applications is not questionable because of improved quality of subsequent motion analysis, as well as expanded data usability. Three major uses of range data are:
1) assistance in correspondence estimation;
2) separation of the active (or manipulating) hand from other body parts or skin-colored objects; and
3) acquisition of the resulting object trajectory in 3-D for motion analysis.

First, the projection of human movement is always affected by the observation viewpoint and the distance from the camera [19]. Since machine vision systems try to recover useful information about a scene from its projections, having three-dimensional (3-D) data eliminates ambiguities in solving the inversion of a many-to-one mapping [5]. Most gesture-tracking and recognition applications will certainly benefit from including range data and recovering additional information from a scene.

The latter also satisfies the second goal, eliminating any reliance on background subtraction techniques and workspace restrictions (color-coding, movement constraints, or limiting initialization assumptions). This is necessary to achieve usability in large visualization or virtual reality environments. Such applications involve dynamic backgrounds that include other people interacting with the system, as well as skin-colored elements in objects of interest and in backgrounds.

The third benefit of range data is an obvious improvement in trajectory analysis. Trajectory analysis is domain-specific, yet most recognition and prediction tools rely on 3-D input. Trajectory determines the type of gesture and other application-specific parameters. They are passed by a motion-interpretation interface passes to the back-end system.

1.2 Problems

Until recently, however, using range data for tracking and other interactive applications was not feasible due to speed and reliability considerations. Range image acquisition and computation is a time-consuming process. For instance, it requires on average more then 30 seconds to acquire and compute range data using a K2T scanner on a SUN SPARC 20, and more than 2 minutes using a Cyberware scanner on a Silicon Graphics O2 (considering higher precision achieved with the latter). Recent availability of less expensive, faster range data (for instance, a Digiclops system [8]) makes it a feasible source of information for interactive HCI applications; however, an additional speedup is still needed. Also, an obvious trade-off results in a significant loss in the quality of data. This necessitates a noise analysis of range data. Obvious approaches, such as choosing median range values for segmented regions or even averaging, lead to ill-defined trajectories that cannot be smoothed or interpolated with conventional techniques. Attempts to improve the efficiency of range computation also face significant difficulties, as selected samples may contain too much noise to be reliable.

[1] This work was performed under the auspices of the U.S. Department of Energy by University of California Lawrence Livermore National Laboratory under contract number W-7405-Eng-48. UCRL-JC-145664

1.3 Previous Work

Some researchers have used multiple cameras and models to obtain 3-D locations of body parts. Azarbayejani and Pentland [2] triangulated on blobs comprising a model. Gavrila and Davis [7] addressed whole-body tracking with four cameras placed in the corners of the room. Segen and Kumar [15] used depth cues from projections of the hand and its shadow for 3-D hand-pose estimation. Davis and Shah presented a tracking method by fitting 3-D models (generalized cylinders) to fingers in a 2-D image [5]. Otherwise, range data was used in motion analysis primarily in an offline mode [17]. Recent approaches to improving efficiency of range data (such as [16]) attempted to limit stereo computation to smaller regions found in respective color images. This paper introduces a "self-reliant" range processing mode in which noise is effectively filtered out, allowing the robust dynamic adjustment of minimum and maximum disparities.

Traditional approaches to tracking typically relied on segmentation of the intensity data, using motion or appearance data. A majority of the methods began by segmenting the human body from the background. For instance, in "blob approaches" people were modeled as a number of blobs resulting from pixel classification based on their color and position in the image. Wren *et al.* [18] achieved segmentation by classifying pixels into one of several models, including a static world and a dynamic user represented by gaussian blobs. Yang and Ahuja [20] used skin color and the geometry of palm and face regions for segmentation stages of their system. A Gaussian mixture (with parameters estimated by an EM algorithm) modeled the distribution of skin-color pixels. Rehg and Kanade [14] used a 3-D hand model to track a hand. They compared line features from the images with the projected model, and performed incremental state corrections. Similar work was presented by Kuch and Huang [10] in which the synthesis process could fit the hand model to any person's hand. Bobick and Wilson [3] treated gesture as a sequence of states and computed configuration states along prototype gestures. Yacoob and Black proposed parameterized representation of human movement [19]. Cutler and Davis [4] segmented the motion and computed a moving object's self-similarity (including human motion experiments). A review by Aggarwal and Cai [1] classified approaches to human motion analysis, the tasks involved, and major areas related to human motion interpretation. A review by Pavlovic *et al.* [12] addressed main components and directions in gesture recognition research for HCI.

1.4 Overview

A representative frame in each image sequence consists of both intensity and range images (Figure 1) showing the execution of basic sets of gestures applicable to visualization or virtual object manipulation (such as zoom, rotation and translation). Problems are notable in the range image.

The premise of this work is that, given noisy and generally unreliable range data, we can filter out erroneous information using range histograms (Section 2.1). Next, using the resulting consistent data that has passed filters, it is possible to limit the depth search space dynamically (Section 2.2). Such localization is possible if expectations of the movement behavior are available. This allows for efficient hand tracking (Section 2.3). An application to virtual reality navigation is presented (Section 3). Experimental results, underlying theory and conclusions are also presented.

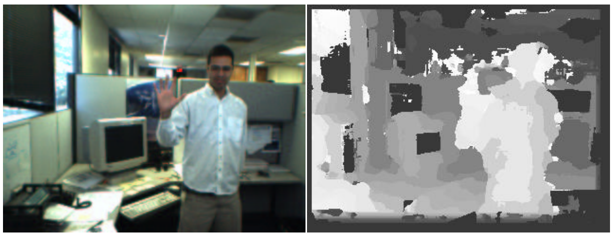


Figure 1: Typical intensity and range images. Note problems in the range data.

2 Solutions

2.1 Histogram-Based Range Filtering

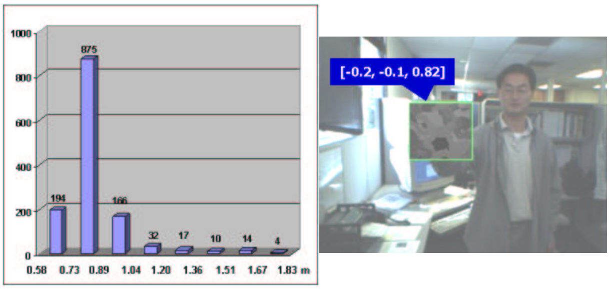
Range data quality is the cental issue. In recent systems it is traded off for some additional speed; and testing is needed before including it in motion estimation. It is obvious (Figure 1) that acquired range is not smooth, and noise is quite significant even in foreground objects. A gesture-tracking system normally includes estimation of 3-D coordinates for a hand(s) used in object manipulation. When a high-precision scanner is used, choosing median range values for segmented regions, or even averaging, is an adequate approximation needed by the trajectory analysis. Experiments have shown, however, that such conventional approaches fail when presented with data sets where the signal to (overall) noise ratio could be as low as 1 for many frames. This necessitates a noise analysis of range data.

It has been observed that correct range values for a small region of interest (ROI), such as a human hand, are clustered together in depth. Noisy depth estimates are distributed across the range of disparities searched. Therefore, a logical solution involves a histogram of range values of skin pixels for each ROI. The scope of computed depth values is split into a number of bins. When the program starts, minimum and maximum disparities are set according to the expected depth of the scene. In our case, a rather large workspace volume is included: from 0.2m to 5.0m. Since each bin is set to 0.16m width, the system starts with 30 bins. The bin width was determined experimentally to cover the depth needed for various hand positions. Then, the system dynamically adjusts the range of disparities to cover [-0.25m, 1.0m] from the previous hand location which results in 8 bins. If needed, the system can vary the limits as well.

A representative summary of selected skin regions for a frame is shown in Table 1. Two regions are analyzed, R_1 (1532 pixels) and R_2 (1312 pixels). This table corresponds to Figure 2. Each bin in Figure 2(a) also represents a range of depth values. The largest bin is assumed to be a representative collection of pixels for the ROI; everything else is considered noise. The importance of histogram-based filtering leads to correct hand detection and tracking (described in details in Section 2.3). In regards to the example shown, computation of "centers of gravity" (COGs) of respective regions yields [0.05, -0.26, 1.37] and [-0.20, -0.06, 0.82]. Since we assume that the region of interest (the hand involved in object manipulation) is the closest skin region to the camera (see Section 2.3), R_2 is identified as the hand region. This is illustrated by overlaying the range data and the COG on Figure 2(b). Analysis of results shows that the technique is very robust and does not slow the process down.

Table 1: Histogram for a frame with a depth range [0.58, 1.83]. Two regions are analyzed, R_1 (1532 pixels) and R_2 (1312 pixels).

Bin	Range	R_1, pix	%	R_2, pix	%
0	[0.58, 0.73]	7	0.46	194	14.79
1	[0.73, 0.89]	28	1.83	875	66.70
2	[0.89, 1.04]	11	0.72	166	12.65
3	[1.04, 1.20]	2	0.13	32	2.44
4	[1.20, 1.36]	1060	69.19	17	1.30
5	[1.36, 1.51]	424	27.68	10	0.76
6	[1.51, 1.67]	0	0	14	1.07
7	[1.67, 1.83]	0	0	4	0.30



(a) (b)

Figure 2: Range histogram (a) and resulting region location estimation (b).

2.2 Dynamic Disparity Localization

Range Computation. The Digiclops stereo vision system [8] computes range based on triangulation between cameras. It consists of a three-camera module. Offset in positions of the cameras produces differences in resulting images. These images are compared using square masks to establish correspondences [8]:

$$\min_{d=d_{min}}^{d_{max}} \sum_{i=-\frac{M}{2}}^{\frac{M}{2}} \sum_{j=-\frac{M}{2}}^{\frac{M}{2}} |I_{right}[x+i][y+j] - I_{left}[x+i+d][y+j]| \quad (1)$$

where d_{min} and d_{max} are the minimum and maximum disparities, M is the mask size, I_{right} and I_{left} are the right and left images, respectively [8]. Since the camera parameters (their relative positions, the focal length and resolution) are fixed, re-calibration is not usually required. According to the multi-baseline stereo theory [11] used in the stereo computation by the system, distance z to the scene point is related to the disparity d, baseline length B and focal length F:

$$z = BF\frac{1}{d} \quad (2)$$

The total amount of computation for stereo processing per frame (required for the Sum of Absolute Differences algorithm) is estimated as [11]:

$$N^2M^2d(C-1)P \quad (3)$$

where N^2 is the image size, C is the number of cameras (three for the system used), and P is the number of operations per one square difference calculation.

Dynamic Adjustment of Disparities. It has therefore been observed that the speed of range computation is affected by a range of disparities considered in the system. Larger range allows us to take more possibilities into account, avoiding generation of an erroneous depth information for a part of the scene. It leads, however, to a drastic slowdown, which is clearly unacceptable in a HCI system. This search depth can be limited if the range is reliable (as discussed in Section 2.1), and if some prior knowledge about the behavior of the ROI is available.

Human behavior during the interaction can be quantified to define such limits. In our applications (gesture-controlled visualization and virtual reality) gestures take more than 1 second, the range of hand motion is less than 1m (an approximate arm length). Therefore, at a rate of 5–6 frames/second (fr/sec), the difference in the hand position between two frames can be between 0.17 and 0.20m (forward or backward). Adopting an even more conservative estimate for robustness, we search +/-0.25m from the previous hand location. However, this range is expanded to include the body of the interacting person, which cannot be more than 1m behind the manipulating hand. Range computation of other skin regions such as a face is necessary for successful differentiation between them. Thus, the resulting range [-0.25m, +1.00m] spans a possible hand movement forward, to the depth where the human body can be found (Figure 3). Spatio-temporal correlation produces a possibility of searching within a smaller region, based on the match in the previous frame.

The system still needs to search the entire scene depth [0.2m, 5.0m] upon the initialization to get its bearings. Another possibility for reset occurs when the depth becomes negative, which has never occurred in our experiments to date. This prevents trapping in the wrong depth range, for instance, when a wrong region is temporarily considered to be a ROI. Tracking is never perfect, and recovery has to be transparent to the system, so that the loss of one frame does not degrade the quality of motion analysis. Frame rates for different sequences on a Pentium 4 PC 1.5GHz with 512 MB RAM are shown in Table 2 (other results for these sequences are discussed in Table 3). The average rate is 5.51 fr/sec, which yields a speedup of 60.2% over experiments with a static disparity range selection [0.2m, 5.0m] (on average 3.44 fr/sec).

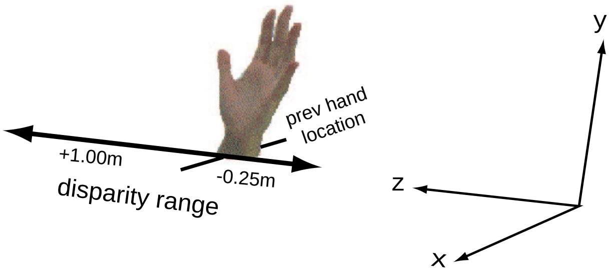


Figure 3: Illustration of dynamic disparity range adjustment.

Table 2: Frame rates (fr/sec) for different sequences.

Seq.	1	2	3	4	5	6	7
Rate	5.57	5.52	5.54	5.49	5.42	5.49	5.57

2.3 Detection of Manipulating Hands

We detect the manipulating hand by detecting skin regions using color and shape, and then differentiating the hand involved in manipulation using its 3-D location. We introduce the SCT color space first.

SCT Color Space. The color image analysis is difficult because color data is highly sensitive to lighting conditions. Stability is obtained when the color attributes of a certain region are the same regardless of lighting conditions. Therefore, many researchers have attempted to transform color to a system that is less influenced by the changes in illumination.

The spherical coordinate (SCT) transform [13] separates illumination information from color information. It transforms [R,G,B] to [L,A,B] by

$$L = \sqrt{R^2 + G^2 + B^2} \tag{4}$$

$$\angle A = cos^{-1}\left[\frac{B}{L}\right] \tag{5}$$

$$\angle B = cos^{-1}\left[\frac{A}{Lsin(\angle A)}\right] \tag{6}$$

L is the distance of the color from the origin of RGB color space. $\angle A$ is the angle that the vector of color from origin makes with the blue axis, while $\angle B$ is the angle that the projection of vector to the RG plane forms with the red axis.

Skin Region Detection in Color Images. Skin pixels are classified with a minimum distance classifier by Mahalanobis distance. First, we train the skin classifier on a set of collected sample skin pixels (offline). During the detection, the [R,G,B] pixel values are transformed to [L,A,B] using SCT. Since L corresponds to the illumination information, we need only A and B to represent the skin color. Each skin pixel forms a vector (X) and we compute the mean vector M_x and covariance matrix C_x. Then, the Mahalanobis distance r is defined as

$$r = \sqrt{(X - M_x)C_x^{-1}(X - M_x)} \tag{7}$$

If $r \leq T_{skin\_max\_dist}$, then X is classified as a skin color. We used 0.75 for $T_{skin\_max\_dist}$. To filter out the small regions (noise), an erosion operation is applied, followed by dilation. The erosion changes the skin pixel to a non-skin pixel if 2 or more of its 4-connected neighbors are background. The dilation converts the non-skin pixel to skin pixel if any of its 4-connected neighbors is a skin pixel. After the noise removal, the skin image is segmented using connected component analysis. Then, the regions smaller than $T_{min\_region\_size}$ are eliminated. The threshold is computed as $0.0012 \cdot R \cdot C$ where the image dimension is $R \times C$. For our experiment with image size of 320×240, the minimum size is 92 pixels. We noted that the rectangular hand regions are usually close to squares. Thus, we discard regions whose height:width is less than $T_{min\_HW\_ratio}$ (meaning regions that are too flat) or greater than $T_{max\_HW\_ratio}$ (meaning that the regions are too tall). We used the range of [0.25, 4.0] for the height:width ratio.

Identification of the Manipulating Hand. The previous step detects skin regions including faces, hands, legs and other body parts, since they all could pass the height:width ratio check. We assume that the hand gesture occurs in front of the body. So, the skin region closest to the camera is identified as the manipulating hand. The 3-D location of each pixel is given by range images. We determine the 3-D location of the region by using histogram-based range filtering as mentioned in Section 2.1. Since the z increases from the camera toward the scene, the region with the smallest z is identified as the manipulating hand region (as shown in the example discussed in

Section 2.1). We noted that the accuracy of range data along the boundary of the skin regions could be especially low. Therefore, we first perform erosion on the skin regions, and then compute the 3-D location of regions using the pixels inside the boundary.

Table 3 includes results for seven sequences used for testing. Of 1043 images, the algorithm correctly identified the manipulating hand 1041 times (99.8%). Two erroneous results are due to identical depth readings obtained for the face and the manipulating hand in those two cases. Selected frames for two sample manipulations (translation and zoom) are shown in Figure 4.

Table 3: Hand detection results for seven sequences (MH=manipulating hand, NMH=non-MH, OSR=other skin regions, NSR=non-skin regions, TMH=total MH.

	correct	incorrect			
dataset	MH	NMH	OSR	NSR	TMH
1	98	0	0	0	98
2	199	0	0	0	199
3	98	0	0	0	98
4	198	0	0	0	198
5	50	0	0	0	50
6	198	0	2	0	200
7	200	0	0	0	200
total	1041	0	2	0	1043
total (%)	99.8	0.0	0.2	0.0	100.0

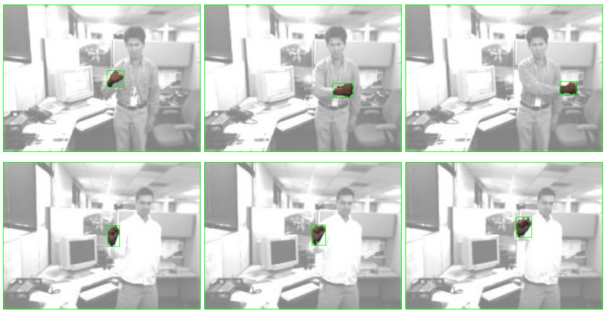


Figure 4: Selected frames for two sample manipulations: translation (top row) and zoom (bottom row). Hands used for manipulation are (automatically) detected and shown in natural skin colors, whereas the rest of each image is displayed as greyscale for visualization.

3 Application to Virtual Reality Navigation

An important part of any virtual reality (VR) system is position-tracking and mapping [6]. Tracking is defined as the real-time position and orientation estimation of a moving object, and mapping refers to surface measurements. Depending on the application, tracking certain body parts is fundamental to its success (for instance, head and eye tracking for visual displays, hand and arm tracking for haptic interfaces, and body surface mapping for videoconferencing). Tracking human body parts is also vital in the context of controlling computer-generated (or remote) objects (for example, in telerobotics).

Most existing trackers suffer from discomfort-related issues [6]. For example, mechanical trackers are inexpensive and reasonably accurate. Body-based trackers (such as hand controllers, joysticks, or helmet attachments), however,restrict spontaneity and natural motion, whereas ground-based devices (e.g., hand controllers) limit the workspace by literally binding an operator to the ground. Another disadvantage is the limited number of degrees of freedom (DOFs) that can be measured (when multiple limbs are moving).

Similarly, traditional controls can also be replaced with gestures (manipulations). Conventional controls include 3- and 6- dimensional mice/trackball/joystick devices. They differ from their desktop equivalents by having extra buttons and wheels that are used to control not just the XY translation of a cursor, but its Z dimension and rotations in all three directions. This section shows that optical trackers/controls can be used instead of, or in addition to, other control and tracking elements (Figure 5). Combination of these functions points to the efficiency of the approach, in addition to the comfort issues already discussed.

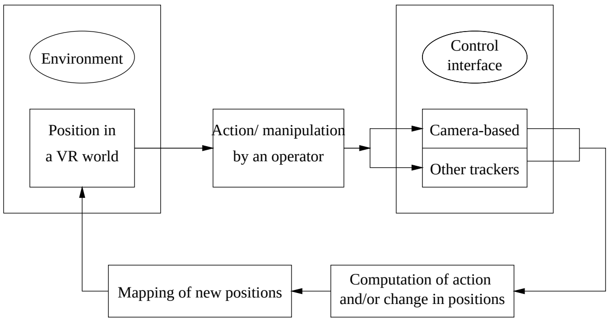


Figure 5: Integration of optical processing into a VR system.

Computed 3-D coordinates of hand positions from tracking zoom and translation gestures (input sequences are not shown; they are similar to ones shown in Figure 4) are transfered to a VR system. Figure 6 shows the application of computed hand-motion to the virtual hand (view from above, the hand is a smaller moving circle with crosshairs). An arrow in the first images represents direction of the motion in the XY plane. Only selected frames are included.

4 Conclusions

The usefulness of range data in interfaces based on human-computer communication applications is not questionable because of improved quality of subsequent motion analysis as well as expanded data usability. Although less costly, range acquisition and pro-

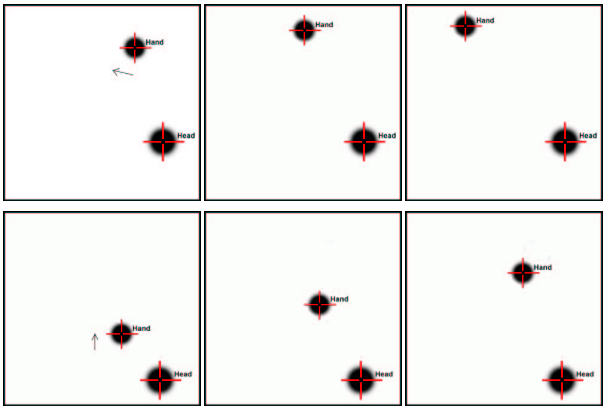


Figure 6: Applying computed hand motion to the virtual hand (view from above) for a zoom (top row) and translation (bottom row) gestures.

cessing still presents a speed vs. data reliability tradeoff. This paper proposed a method that, given noisy and generally unreliable range data, can filter out erroneous information using range histograms. Next, using the resulting consistent data that has passed filters, this method limited the depth search space dynamically using motion history and its current state. With fast and reliable range data, an object of interest can be separated from other objects or background by depth alone. The algorithm also includes a robust skin-color segmentation that is insensitive to changes in lighting conditions.

Experimental results demonstrated the success of the proposed algorithm. Using filtered range data, the algorithm correctly identified the hand involved in manipulation 99.8% of the time. Dynamic disparity adjustment produced a speedup of 60.2% over a static disparity range selection. An application to virtual reality navigation was also presented.

Other applications tracking the motion of the human body can benefit from improved quality and speed of range data. These applications include videosurveillance, gesture-based interfaces for multimedia applications and systems, interfaces for people with disabilities that prevent them from using the standard input technology, and videoconferencing.

References

[1] J. K. Aggarwal and Q. Cai. Human motion analysis: A review. In *IEEE Nonrigid and Articulated Motion Workshop*, pp. 90–102, San Juan, PR, June 1997.

[2] A. Azarbayejani and A. Pentland. Real-time self-calibrating stereo person tracking using 3-D shape estimation from blob features. In *Proc. ICPR*, Vienna, August 1996.

[3] A. F. Bobick and A. D. Wilson. A state-based approach to the representation and recognition of gesture. *IEEE Trans. on PAMI*, 19(12):1325–1337, December 1997.

[4] R. Cutler and L. Davis. Real-time periodic motion detection, analysis, and applications. In *Proc. CVPR*, vol. 2, pp. 326–332, Fort Collins, CO, June 1999.

[5] J. Davis and M. Shah. Toward 3-D gesture recognition. *Intern. J. of Pattern Recognition and Artificial Intelligence*, 13(3):381–388, May 1999.

[6] N. I. Durlach and A. S. Mavor, editors. *Virtual Reality: Scientific and Technological Challenges.* National Academy Press, Washington, DC, 1995.

[7] D. Gavrila and L. Davis. 3-D model-based tracking of humans in action: a multi-view approach. In *Proc. CVPR*, pp. 73–80, San Francisco, CA, June 1996.

[8] Point Grey Research Inc. *Triclops Stereo Vision System Version 2.1, User's guide and command reference.* Inc., Point Grey Research, Vancouver, BC, 1996.

[9] T. Kanade, A. Yoshida, K. Oda, H. Kano, and M. Tanaka. A stereo machine for video-rate dense depth mapping and its new applications. In *Proc. CVPR*, pp. 196–202, June 1996.

[10] J. J. Kuch and T.S. Huang. Model-based tracking of self-occluding articulated objects. In *Vision based hand modeling and tracking for virtual teleconferencing and telecollaboration*, pp. 666–671, Cambridge, MA, June 1995.

[11] K. Oda, M. Tanaka, A. Yoshida, H. Kano, and T. Kanade. A video-rate stereo machine and its application to virtual reality. In *Proc. ISPRS '96*, 1999.

[12] V. I. Pavlovic, R. Sharma, and T. S. Huang. Visual interpretation of hand gestures for human-computer interaction: A review. *IEEE Trans. on PAMI*, 19(7):677–695, July 1997.

[13] M. W. Powell and R. Murphy. Position estimation of micro-rovers using a spherical coordinate transform color segmenter. In *Proc. of IEEE Workshop on Photometric Modeling for Computer Vision and Graphics*, pp. 21–27, Fort Collins, CO, June 1999.

[14] J. M. Rehg and T. Kanade. Visual tracking of high DOF articulated structures: an application to human hand tracking. *Proc. ECCV*, 2:35–46, May 1994.

[15] J. Segen and S. Kumar. Shadow gestures: 3D hand pose estimation using a single camera. In *Proc. CVPR*, vol. 1, pp. 479–485, Fort Collins, CO, June 1999.

[16] L. V. Tsap. Real-time local range on-demand for tracking gestures. In *Proc. IEEE Workshop on Human Modeling, Analysis and Synthesis*, pp. 52–58, Hilton Head Island, SC, June 2000.

[17] L. V. Tsap, D. B. Goldgof, and S. Sarkar. Nonrigid motion analysis based on dynamic refinement of finite element models. *IEEE Trans. on PAMI*, 22(5):526–543, May 2000.

[18] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Trans. on PAMI*, 19(7):780–785, July 1997.

[19] Y. Yacoob and M. J. Black. Parameterized modeling and recognition of activities. *J. of Computer Vision and Image Understanding*, 73(2):232–247, 1999.

[20] M.-H. Yang and N. Ahuja. Recognizing hand gestures using motion trajectories. In *Proc. CVPR*, vol. 1, pp. 466–472, Fort Collins, CO, June 1999.